ESD-TR-68-213

A COMPUTER PROGRAM FOR BACKSCATTER BY SMOOTHLY JOINED, SECOND DEGREE SURFACES OF REVOLUTION - 2430-6

C.E. Ryan, Jr.

April 1968

DEPUTY FOR SURVEILLANCE AND CONTROL SYSTEMS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

1 of 2
ESSXS

(Prepared for Contract No. AF 19(628)-67-C-0308 by The Ohio State University, ElectroScience Laboratory, Department of Electrical Engineering, 1320 Kinnear Road, Columbus, Ohio.)

LEGAL NOTICE

OTHER NOTICES

ABSTRACT

A computer program for calculation of the echo area of smoothly joined, N section convex conducting surfaces of revolution, described by a second degree equation is presented. For the case of E_θ (parallel) polarization of the incident and scattered fields the solution is obtained by a combination of geometrical optics and creeping wave theory. For the case of E_ϕ (perpendicular) polarization the solution is obtained using geometrical optics, and the creeping wave is neglected. The computed results for E_θ polarization are in good agreement with measurements for prolate spheroids, prolate spheroid-sphere, and prolate spheroid-oblate spheroid combinations.

TABLE OF CONTENTS

A COMPUTER PROGRAM FOR BACKSCATTER BY SMOOTHLY JOINED, SECOND DEGREE SURFACES OF REVOLUTION

I. INTRODUCTION

The computer program listed in Appendix I, applies the geometrical optics and creeping wave solutions described in Ref. 1, 2 to obtain the backscattered fields of a surface of revolution. The target is composed of N sections, each section described by a second degree equation. In addition the target must be convex, and be smoothly joined at the boundaries between sections. The program as listed computes the backscattered field and echo area for E_θ (parallel) polarization as a function of the incidence angle, for a given wavelength. The geometrical optics backscattered field is independent of polarization, thus the geometrical optics scattered field for the case of E_ϕ (perpendicular) polarization is also obtained. The creeping wave scattered field has not been included for E_ϕ polarization due to the difficulty in obtaining the ray path of the creeping wave in this case.[1] This program has been tested for E_θ polarization for prolate spheroid, prolate spheroid-sphere, and prolate spheroid-oblate spheroid targets. The results of these tests are presented in Ref. 1.

II. TARGET DESCRIPTION

The surface is described in each section by the second degree equation

$$(1) \qquad F(r,\theta) = A_1\, r^2 \sin^2\theta + A_2\, r^2 \cos^2\theta + A_3 r^2 \sin\theta\cos\theta + A_4\, r\cos\theta + A_5\, r\sin\theta + A_6 = 0 \quad .$$

The constants $A_1 \ldots A_6$ specify the surface in the section (i) bounded by the angles θ_i and θ_{i+1}. The surface which may be represented by Eq. (1) includes the sphere, prolate and oblate spheroids, and the ogive. In general any surface derived from a conic section can be described by this equation. More complex surfaces may be represented by using a large number of sections and approximating the desired surface by a second degree surface within each section. This program provides for 20 sections but this number can be readily increased.

A restriction on the target specification, which is a result of the method for finding the specular point, is that the coordinate origin be located such that the normal to the surface at $z = 0$ be r directed. That is, $dr/d\theta|_{z=0} = 0$. Discontinuities in the derivatives $(d^n r/d\theta^n)$ at the junctions between sections may exist. However, the effects of such discontinuities as well as the effects of tips are not included in this program. A discontinuity in the first derivative $(dr/d\theta)$ at the junction ("wedge" discontinuities) causes diffraction, and can be evaluated using wedge diffraction techniques.[1,2] A discontinuity in the second derivative $(d^2 r/d\theta^2)$ at the junction results in reflection and transmission of an incident creeping wave at such a discontinuity. An example of this effect, the spherically capped ogive, has been discussed in Refs. 1 and 2. As the effects of a discontinuity in the first derivative are significent this program may not give good results for such a target. The creeping wave reflection effect due to a moderate second derivative discontinuity may be neglected with good results.[1] Thus good results for the scattered field of perfectly conducting convex, smoothly joined surfaces of revolution may be obtained using this program.

III. THE CREEPING WAVE COMPUTER PROGRAM

The creeping wave computer program given in Appendix I uses geometrical optics and creeping wave theory to calculate the backscattered field of the target. The geometrical optics field is obtained by identifying the specular point and calculating the Gaussian curvature at the specular point. The backscattered field is then calculated as given in Ref. 1. The creeping wave backscattered field is obtained by identifying the points of attachment and reradiation of the creeping wave, calculating the diffraction coefficient at these points, and by computing the attenuation of the creeping wave on a path defined by the E-plane of the target. The backscattered field is then calculated as given in Ref. 1. These contributions are added to obtain the total backscattered field.

Referring to the computer program listing shown in Appendix I, the function of the significant sections of the program will be discussed. The card numbers associated with each section will be specified. This discussion, together with the comment cards included in the program listing, is intended to give sufficient information about the program to enable a qualified programmer to both use and modify the program. Statements which are in common use in Fortran IV, such as DIMENSION, COMPLEX, and FORMAT statements will not be discussed as it is assumed that the reader has a knowledge of Fortran IV.

The COMMON declaration (0011) is used to store the constants required in Eq. (1) in the common block labelled /DATA/. This common block is used in conjunction with the unlabelled common block to transfer a particular set of constants A1 (I) to A6 (I) into the unlabelled common regions shared by the subroutines. This provision reduces the number of calling variables required by each subroutine.

The COMMON declaration (0013) is used to store and manipulate the angles corresponding to the section boundaries.

The statements 0018-0022 initialize constants which are required in the calculations.

The READ statements 0023-0027 read the required data, and provision is also made to write out this data for the purpose of identification.

The statements 0028-0030 set up the incrementation of the incidence angle THT and the statement 0031 calculates the propagation factor FK.

Next the loop for incrementing the incidence angle is entered, and the angle converted to radians. The logic statements (0036, 0037) serve two purposes if the incidence angle exceeds 90°. First the incidence angle is constrained to the range $0. < \text{THT} < \pi/2$. Then the subroutine FNVRT (N) is called. The subroutine FNVRT (N) inverts the target, i.e., the portion of the target defined as section #1 becomes section #N, section #2 becomes section #(N-1) and so forth. This provision is necessary so that the propagation direction of the creeping waves which start at the point of attachment is always the positive angular direction. This provision simplifies the logic required to compute the creeping wave contributions. Figure 1 illustrates this provision.

The propagation vector and polarization vector of the incident wave are computed next (0038-0046). These vectors are used in the determination of the specular point and the points of attachment and reradiation of the creeping wave.

Next, the location of the specular point is determined (0047-0060). This is done by incrementing by DTHB along the surface of the target in the $\phi = 0$ plane, performing the scalar product of the propagation vector and the surface normal vector at each point, and by finding the point at which this product is a maximum. This point, in section NSP, is the specular point (THSPP).

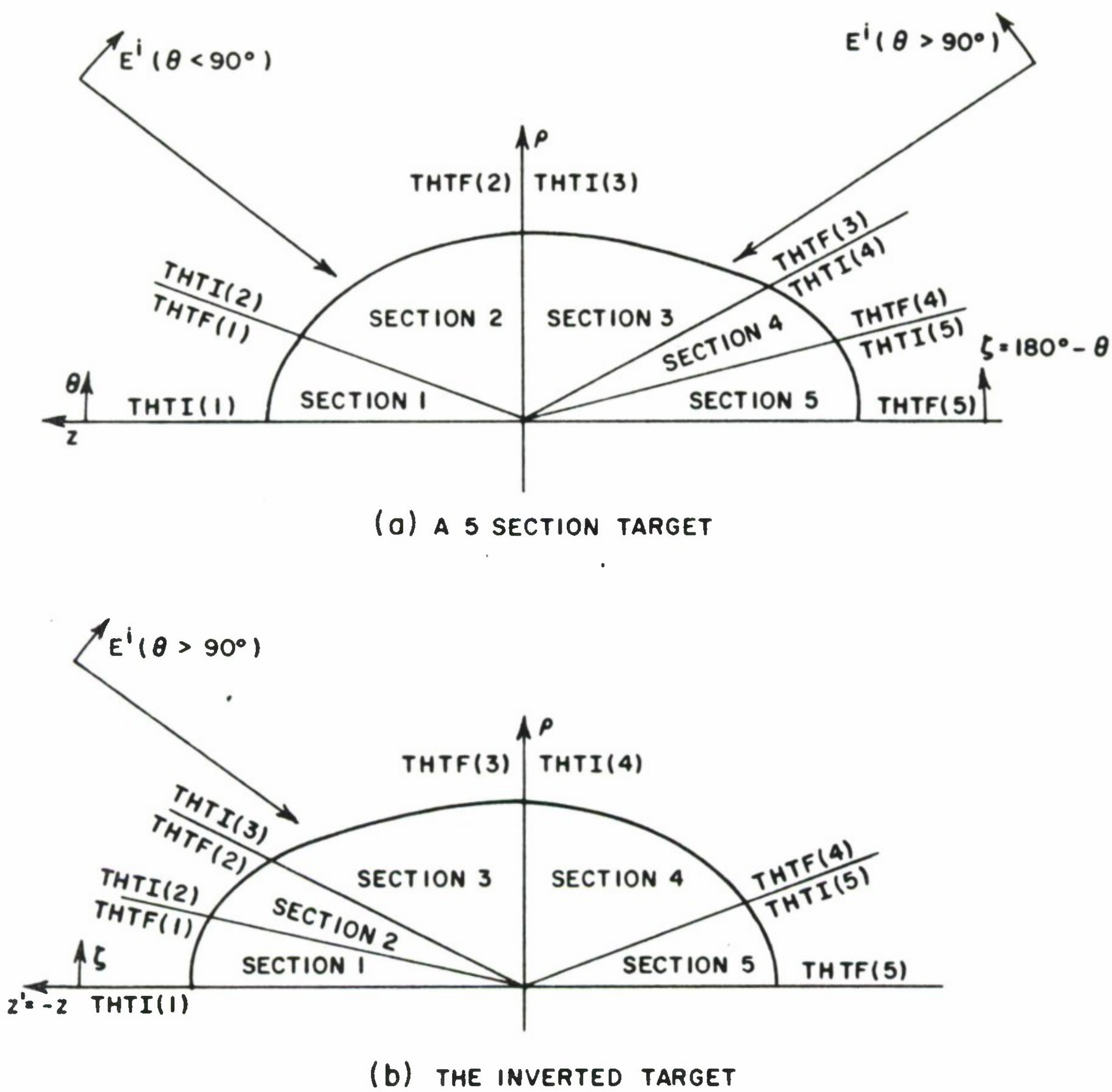


Fig. 1. A target (a) and the inversion (b).

The location of the points of attachment and reradiation of the creeping wave for $\phi = 0, \pi$ is determined next (0062-0090). This is done in the same manner as the determination of the specular point location except that the scalar product of the polarization and normal vectors is used. After these points have been found they are written out together with the specular point (0091, 0092).

Having determined the location of the specular point (NSP, THSPP, RSP) the Gaussian curvature at the specular point is computed and used to calculate the geometrical optics scattered field (0093-0103).

The creeping wave path length is computed in 0104-0129. This is done by starting at the attachment point (THCWL, $\phi = 0$) and performing a numerical integration of the differential arc length along the surface to the point $\theta = \pi$. The length thus determined is CWL1. This process is then repeated for the attachment point (THCWL, $\phi = \pi$) with a result CWL2. The product of the propagation factor and the sum of the path lengths then gives the free space phase of the creeping wave which propagates around the target.

Next the complex attenuation of the creeping wave is computed (0130-0150). The same process used to compute the path length is applied, with the exception that the complex attenuation coefficient must be integrated. This is accomplished using the EXTERNAL ALPHDS as the integration function.

Having obtained the creeping wave path length, attenuation, and points of attachment and reradiation, it is easy to calculate the creeping wave scattered field. This is done in 0151-0161. The phase is first calculated and the square of the diffraction coefficient determined. Then the creeping wave scattered field ECW is computed.

The total scattered field and echo area in square wavelengths are computed in 0162-0165. Next the common regions are reset if THTD $> 90^{\circ}$ and the incidence angle in degrees, the geometrical optics, creeping wave, total scattered field and echo area are written. At the completion of the loop (0169) the program is terminated.

IV. FUNCTIONS AND SUBROUTINES

In addition to the arguments required to call each subroutine or function as given in the description below, it is necessary to transfer the constants in Eq. (1) for the section (I) from the /DATA/ common block to the unlabelled common block. All function and subroutines in which these COMMON statements appear require that such a transfer be made. This provision reduces the number of arguments required in the functions and subroutines.

Complex Function DELSP(THT) 0172-0179

This function calculates the incremental arc length in the θ direction at THT. Although the arc length is a real number this function is declared complex so that it may be used as an external function in the numerical integration.

Subroutine FNVRT(N) 0180-0195

This subroutine interchanges the data which specifies the target, thus inverting the target by 180° in θ.

Complex Function SPHASE (THTI, PHII, THTB, PHIB, RB, FK) 0196-0204

This function determines the phase of the incident field (THTI, PHII) at a point on the target (RB, THTB, PHIB) for a propagation factor FK.

Complex Function PHASE (THTI, PHII, THTB, PHIB, RB, FK) 0205-0214

This function determines the backscattered phase of the field at (RB, THTB, PHIB) for an incident field (THTI, PHII) and propagation factor FK.

Complex Function ALPH (RH01, RH02, WAVE) 0215-0230

This function computes the complex creeping wave attenuation coefficient for the orthogonal radii of curvature RH01 and RH02 and a wavelength WAVE. RH01 is the radius of curvature in the propagation direction.

Complex Function DSQ (RI1, RSI, WAVE) 0231-0243

This function computes the square of the creeping wave diffraction coefficient as a function of the radii of curvature RI1, RS1 in the propagation direction and the wavelength WAVE.

Complex Function ALPHDS (THT) 0244-0260

This function determines the product of the creeping wave attenuation coefficient and the metric of the surface which is needed in the integration for the attenuation along the path.

Subroutine DIFFGO (R, THT, RTHT, RTTH, ECAP, FCAP, GCAP, ELC, FLC, GLC) 0261-0301

This subroutine calculates the first (RTHT) and second (RTTH) derivatives of the distance (R) from the origin to the surface at the point (R, THT). In addition the coefficients of the first and second fundamental forms of differential geometry of the surface ECAP, FCAP, GCAP, ELC, FLC, GLC are computed.[1]

Subroutine FSPDT (RSP, THSP, FNT, THTS, THTF, DTHT, PHI, VX, VY, VZ) 0302-0326

This subroutine finds the point (RSP, THSP) in the section FNT bounded by THTS and THTF where the scalar product of the surface normal vector and the vector (VX, VY, VZ) is a maximum. This is done by incrementing in angle by DTHT and selecting the largest scalar product.

Subroutine FINT (SSS, FCTI, FLL, FUL, ERRR, NX) 0327-0377

This is a subroutine for numerical integration of the complex external function FCTI between the limits FLL and FUL. The complex result is returned in SSS. ERRR specifies the percent error desired and the integer NX determines whether equal integration increments (NX = 1) or adjusted increments (NX = 2) are used. A description of this integration technique is given in Ref. 3.

Subroutine FNORM (FNVX, FNVY, FNVZ, R, THT, PHI) 0378-0400

The subroutine FNORM calculates the surface normal vector (FNVX, FNVY, FNVZ) at the point on the surface described by the spherical coordinates R, THT, PHI.

Function RAD(THT) 0401-0419

This function computes the distance from the origin to the point on the surface at the angle THT.

Subroutine FCOMM (I) 0420-0431

This subroutine transfers the constants required by Eq. (1) for the I^{th} section from the /DATA/ common storage block to the unlabelled common block.

V. DATA

A set of input data is shown in Fig. 2. The order of cards is as follows:

Card 1 - specifies the number of sections (N) in the target.
Card 2 - specifies the wavelength and the increment in incidence angle.
Next N cards - specify the initial and final angular boundaries of the section in radians and the constants required in Eq. (1).

The particular target specified by this data is the prolate spheroid-oblate spheroid combination described in Ref. 1.

```
00002
  1.0          2.0
  0.          1.570796    2.4799     .5102        0.          0.          0.
1.570796    3.1415927   2.4799     5.0955       0.          0.          0.
```

Fig. 2. The input data.

VI. CONCLUSION

A computer program for backscatter by smoothly joined, second degree surfaces of revolution has been developed using the theory presented in Refs. 1 and 2. This program has been tested for prolate spheroids, prolate spheroid-sphere, and prolate spheroid-oblate spheroid combinations with good results.[1] The description of the computer program given in this report should enable a capable programmer to use and modify this program.

The creeping wave computer program was originally intended to be integrated with the computer program based upon wedge diffraction techniques.[1,4] This goal can be obtained by modification of the logic sections of the two programs, and by converting the computational sections of each of the programs to a subroutine form. Thus a program utilizing both wedge diffraction and creeping wave theory can be assembled.

APPENDIX I
THE COMPUTER PROGRAM LISTING

```
$EXECUTE        IBJOB
$IBJOB          GO,MAP
$IBFTC CREEP    LIST,NODECK
  1     FORMAT(1I5)
 2      FORMAT(2F10.5)
  3     FORMAT(8F10.5)
 4      FORMAT(3F15.8)
 5      FORMAT(7F15.8)
   6    FORMAT(7H THSPP=F15.8,GH THCWU=F15.8,7H THCWL=F15.8)
    7   FORMAT(5H NSP=I15,6H NSWU=I15,6H NCWL=I15)
  8     FORMAT(6H CWL1=F15.8,6H CWL2=F15.8)
        COMMONRA1,RA3,RB1,RA9,RA10,RA11,WAVE/DATA/AR1(20),AR3(20),BR1(20),
       CAR9(20),AR10(20),AR11(20)
         COMMON/ANGL/THT1(20),THTF(20),THTP(20)
        COMPLEX PALP1,PALP2
        COMPLEX PHASP,EGO,ALPHDS,ATTEN,ALP1,ALP2,PHCW1,PHCW2,DSQC,ECW,ETOT
        COMPLEX CPTHL,PHASE,ALPH,DSQ,DELSP,CML1,CML2,CEXP
        COMPLEX SPHASE
        DEGRAD=0.01745329
        RADEG=57.29578
        PI=3.1415927
         PI2=PI/2.
        TP=2.*PI
        READ(5,1) N
        READ(5,2)WAVE,DTHT
        WRITE(6,2)WAVE,DTHT
        READ(5,3)(THTI(IR),THTF(IR),AR1(IR),AR3(IR),BR1(IR),AR9(IR),AR10(I
       CR),AR11(IR),IR=1,N)
        DTHB=DEGRAD
        FNT=180./DTHT
        NT=FNT-1.
        FK=TP/WAVE
        DO 100 NTH=1,NT,1
        FNTH=NTH
        THTD=FNTH*DTHT
        THT=DEGRAD*THTD
        IF(THTD.GT.90.) THT=PI-THT
        IF(THTD.GT.90.) CALL FNVRT(N)
C       CALCULATE INCIDENCE VECTOR
        PHI=0.
        VIX=-SIN(THT)
        VIZ=-COS(THT)
        VIY=0.
C       CALCULATE INCIDENT E-VECTOR-PARALLEL POL.
        EIX=COS(THT)
        EIY=0.
        EIZ=-SIN(THT)
C       DETERMINE SPECULAR POINT
        FNSPP=0.
        NSP=0
        RSPP=0.
        THSPP=0.
        D0410 ISP=I,N,1
        CALL FCOMM(ISP)
        CALL FSPDT(RSP,THSP,FNSP,THT1(ISP),THTF(ISP),DTHB,0.,VIX,VIY,V1Z)
        IF(FNSPP.GT.FNSP.OR.THSP.GT.PI2)GOTO 411
        FNSPP=ABS(FNSP)
        RSPP=RSP
        THSPP=THSP
        NSP=ISP
 411    CONTINUE
 410    CONTINUE
C       DETERMINE CREEPING WAVE POINTS
        FCWU=0.
```

```
      NCWU=0
      RSCWU=0.
      THCWU=0.
      DO 510 ICW=1,N,1
      CALL FCOMM(ICW)
      CALL FSPDT(RCWU,THWU,FWU,THTI(ICW),THTF(ICW),DTHB,0.,EIX,EIY,EIZ)
      IF(FCWU.GT.FWU.OR.THWU.LT.PI2)GO TO 511
      FCWU=ABS(FWU)
      RSCWU=RCWU
      THCWU=THWU
      NCWU=ICW
  511 CONTINUE
  510 CONTINUE
      FCWL=0.
      NCWL=0
      RSCWL=0.
      THCWL=0.
      DO 520 ICW=1,N,1
      CALL FCOMM(ICW)
      CALL FSPDT(RCWL,THWL,FWL,THTI(ICW),THTF(ICW),DTHB,PI,EIX,EIY,EIZ)
      IF(FCWL.GT.FWL.OR.THWL.GT.PI2) GO TO 521
      FCWL=FWL
      RSCWL=RCWL
      THCWL=THWL
      NCWL=ICW
  521 CONTINUE
  520 CONTINUE
      WRITE(6,7) NSP,NCWU,NCWL
      WRITE(6,6)THSPP,THCWU,THCWL
C     CALCULATE GEOMETRICAL OPTICS TERM
      CALL FCOMM(NSP)
      CALL DIFFGO(RSPP,THSPP,RDUM,RDDM,ECAP,FCAP,GCAP,ELC,FLC,GLC)
      IF(GCAP) 20,21,20
   21 GAUSS=ELC/ECAP
      GAUSS=GAUSS*GAUSS
      GO TO 22
   20 GAUSS=(ELC*GLC-FLC*FLC)/(ECAP*GCAP-FCAP*FCAP)
   22 CONTINUE
      PHASP=PHASE(THT,0.,THSPP,0.,RSPP,FK)
      EGO=-SQRT(1./GAUSS)*PHASP/2.
C     CALCULATE CREEPING WAVE PATH LENGTH
      EXTERNAL DELSP
      CWL1=0.
      DO 522 NCPU=NCWU,N,1
      CALL FCOMM(NCPU)
      TCW1=THCWU
      TCW2=THTF(NCPU)
      IF(NCPU.GT.NCWU) TCW1=THTI(NCPU)
      CALL FINT(CML1,DELSP,TCW1,TCW2,5.0,2)
      RCML1=REAL(CML1)
      CWL1=CWL1+RCML1
  522 CONTINUE
      CWL2=0.
      DO 524 NCPL=NCWL,N,1
      CALL FCOMM(NCPL)
      TCW1=THCWL
      TCW2=THTF(NCPU)
      IF(NCPL.GT.NCWL) TCW1=THTI(NCPL)
      CALL FINT(CML2,DELSP,TCW1,TCW2,5.0,2)
      RCML2=REAL(CML2)
      CWL2=CLW2+RCML2
  524 CONTINUE
      WRITE(6,8) CWL1,CWL2
      FKL1=FK*CWL1
```

```
      FKL2=FK*CWL2
      FKLCW=FKL1+FKL2
C     CALCULATE CREEPING WAVE ATTENUATION
      EXTERNAL ALPHDS
      ALP1=(0.,0.)
      DO 530 NCPU=NCWU,N,1
      CALL FCOMM(NCPU)
      TCW1=THCWU
      TCW2=THTF(NCPU)
      IF(NCPU.GT.NCWU) TCW1=THTI(NCPU)
      CALL FINT(PALP1,ALPHDS,TCW1,TCW2,5.0,2)
      ALP1=ALP1+PALP1
  530 CONTINUE
      ALP2=(0.,0.)
      DO 532 NCPL=NCWL,N,1
      CALL FCOMM(NCPL)
      TCW1=THCWL
      TCW2=THTF(NCPL)
      IF(NCPL.GT.NCWL) TCW1=THTI(NCPL)
      CALL FINT(PALP2,ALPHDS,TCW1,TCW2,5.0,2)
      ALP2=ALP2+PALP2
  532 CONTINUE
      ATTEN=ALP1+ALP2
C    CALCULATE CREEPING WAVE
      PHCW1=SPHASE(THT,0.,THCWU,0.,RSCWU,FK)
      PHCW2=SPHASE(THT,0.,THCWL,0.,RSCWL,FK)
      CALLDIFFGO(RSCWU,THCWU,RDUM,RDDM,ECAP1,FCAP1,GCAP1,ELC1,FLC1,GLC1)
      CALLDIFFGO(RSCWL,THCWL,RDUM,RDDM,ECAP2,FCAP2,GCAP2,ELC2,FLC2,GLC2)
      FKAP1=ELC1/ECAP1
      FKAP2=ELC2/ECAP2
      RHCW1=1./FKAP1
      RHCW2=1./FKAP2
      DSQC=DSQ(RHCW1,RHCW2,WAVE)
      ECW=-2.*DSQC*PHCW1*PHCW2*CEXP(-ATTEN+(0.,-1.)*FKLCW)
      ETOT=EGO+ECW
      EMAG=CABS(ETOT)
      SIGMA=2.*TP*EMAG*EMAG
      SIGMAL=10.*ALOG10(SIGMA)
      IF(THTD.GT.90.) CALL FNVRT(N)
      WRITE(6,5)THTD,EGO,ECW,ETOT
      WRITE(6,4)THTD,SIGMA,SIGMAL
  100 CONTINUE
      STOP
      END
$IBFTC DELSP.  LIST
      COMPLEX FUNCTION DELSP(THT)
      R=RAD(THT)
      CALL DIFFGO(R,THT,RDUM,RDDM,ECAP,FCAP,GCAP,ELC,FLC,GLC)
      ECAP=ABS(ECAP)
      DELSP=CMPLX(SQRT(ECAP),0.)
      RETURN
      END
$IBFTC FNVRT.  LIST
      SUBROUTINE FNVRT(N)
      COMMONRA1,RA3,RB1,RA9,RA10,RA11,WAVE/DATA/AR1(20),AR3(20),BR1(20),
     CAR9(20),AR10(20),AR11(20)
       COMMON/ANGL/THTI(20),THTF(20),THTP(20)
      PI=3.1415927
       DO 10 I=1,N,1
      BR1(I)=-BR1(I)
      AR9(I)=-AR9(I)
      THTP(I)=PI-THTF(I)
      THTF(I)=PI-THTI(I)
      THTI(I)=THTP(I)
```

```
 10   CONTINUE
      RETURN
      END
$IBFTC SPHAS    LIST
      COMPLEX FUNCTION SPHASE(THTI,PHII,THTB,PHIB,RB,FK)
      FS=1.
      TEST=ABS(PHII-PHIB)
      IF(TEST.EQ.0.) FS=-1.
      FL=RB*COS(THTI+FS*THTB)
      FKL=FK*FL
      SPHASE=CMPLX(COS(FKL),SIN(FKL))
       RETURN
      END
$IBFTC PHAS.    LIST
       COMPLEX FUNCTION PHASE(THTI,PHII,THTB,PHIB,RB,FK)
      FS=1.
      TEST=ABS(PHII-PHIB)
      IF(TEST.EQ.0.) FS=-1.
      FL=2.*RB*COS(THTI+FS*THTB)
      FKL=FK*FL
      PHASE=CMPLX(COS(FKL),SIN(FKL))
       RETURN
      END
$IBFTC ALPH.    LIST
      COMPLEX FUNCTION ALPH(RHO1,RHO2,WAVE)
      COMPLEX ALPH,EX
      EX=CMPLX(0.86603,0.5)
      IF(RHO2.EQ.0.) GO TO 10
      RA=RHO1/RHO2
      U2=(1.48/EXP(0.84*RA))+0.20
      GO TO 11
 10   U2=0.20
 11   AR1=2.*ALOG(RHO1)/3.
      FR1=1./EXP(AR1)
      WV1=ALOG(WAVE)/3.
      FWV=1./EXP(WV1)
      ALPH=U2*FR1*FWV*EX
      RETURN
      END
$IBFTC DSQ.     LIST
      COMPLEX FUNCTION DSQ(RI1,RS1,WAVE)
      COMPLEX DSQ,EX
      EX=CMPLX(.96593,-.25882)
      U1=0.270
      A=SQRT(RI1*RS1)
      AL=ALOG(A)/3.
      WV=2.*ALOG(WAVE)/3.
      FA=EXP(AL)
      FW=EXP(WV)
      DSQ=U1*FA*FW*EX
      RETURN
      END
$IBFTC ALPDS.   LIST
      COMPLEX FUNCTION ALPHDS(THT)
      COMMON AR1,AR3,BR1,AR9,AR10,AR11,WAVE
      COMPLEX ALPHDS,ALP,ALPH
      PI=3.1415927
      R=RAD(THT)
      CALL DIFFGO(R,THT,RTHT,RDDM,ECAP,FCAP,GCAP,ELC,FLC,GLC)
      FMET=SQRT(ECAP)
      RHO1=ECAP/ELC
      RHO2=GCAP/GLC
      IF(THT.EQ.0..OR.THT.EQ.PI) RHO2=RHO1
      RHO1=ABS(RHO1)
```

```
      RHO2=ABS(RHO2)
      ALP=ALPH(RHO1,RHO2,WAVE)
      ALPHDS=ALP*FMET
      RETURN
      END
$IBFTC DIFGO    LIST
      SUBROUTINE DIFFGO(R,THT,RTHT,RTTH,ECAP,FCAP,GCAP,ELC,FLC,GLC)
       COMMON A1,A3,B1,A9,A10,A11,WAVE
C     DIFFERENTIAL GEOMETRY PROPERTIES OF QUADRIC SURFACE OF REVOLUTION
C     F(R,THT)=A1*R*R*SIN(THT)*SIN(THT)+A3*R*R*COS(THT)*COS(THT)+B1*R*R*
C     SIN(THT)*COS(THT)+A9*R*COS(THT)+A10*R*SIN(THT)+A11=0
C     THETA-PHI COORDINATES
C     UPPER AND LOWER CASE F = 0
C     INPUT A1,A3,B1,A9,A10,A11,R,THT
      PHI=0.
      ST=SIN(THT)
      CT=COS(THT)
      S2T=SIN(2.*THT)
      C2T=COS(2.*THT)
      ST2=ST*ST
      CT2=CT*CT
      SP = SIN(PHI)
      CP = COS(PHI)
      U=A1*ST2+A3*CT2+B1*ST*CT
      V=A9*CT+A10*ST
      UTHT=A1*S2T-A3*S2T+B1*(CT2-ST2)
      VTHT=-A9*ST+A10*CT
      UTTH=2.*(A1-A3)*C2T-2.*B1*S2T
      VTTH=-A9*CT-A10*ST
      S1=2.*R*U+V
      S2=R*UTHT+VTHT
      FMAG=SQRT(S1*S1+S2*S2)
      FMAG2=FMAG*FMAG
      FMAG3=FMAG2*FMAG
C     CALCULATE UPPER CASE E,F,G
      RTHT=-R*S2/S1
      ECAP = R*R+RTHT*RTHT
      FCAP = 0.
      GCAP = R*R*ST2
C     CALCULATE LOWER CASE E,F,G
      RTTH=-(2.*RTHT*(2.*R*UTHT+VTHT)+2.*RTHT*RTHT*U+R*(R*UTTH+VTTH))/S1
      ELC=(RTTH*S1+2.*RTHT*S2-R*S1)/FMAG
      FLC = 0.
      GLC=-R*(S1*ST2+S2*ST*CT)/FMAG
      RETURN
      END
$IBFTC FSPDT.  LIST
      SUBROUTINE FSPDT(RSP,THSP,FNT,THTS,THTF,DTHT,PHI,VX,VY,VZ)
C     INPUT THTS,THTF,DTHT,PHI,VX,VY,VZ
      COMMON AR1,AR3,BR1,AR9,AR10,AR11,WAVE
C     SEARCH FOR LARGEST SCALAR PRODUCT IN SECTION
      FND=ABS((THTF-THTS)/DTHT)
      ND=FND+1.
      FNT=0.
      RSP=0.
      THSP=0.
       DO 100 I=1,ND,1
      FI=I-1
      THT=FI*DTHT+THTS
      R=RAD(THT)
      CALL FNORM(FNX,FNY,FNZ,R,THT,PHI)
      FNE=VX*FNX+VY*FNY+VZ*FNZ
      FNE=ABS(FNE)
      IF(FNT.GT.FNE) GO TO 11
```

```
      FNT=ABS(FNE)
      RSP=R
      THSP=THT
 11   CONTINUE
 100  CONTINUE
      RETURN
      END
$IBFTC FINT.    LIST
      SUBROUTINE FINT(SSS,FCTI,FLL,FUL,ERRR,NX)
C     INPUT FCTI,FLL,FUL,ERRR,NX
C     EXTERNAL DECLARATION FOR FUNCTION FCTI REQUIRED
      COMPLEX SSS,SS,FSS,FCTI,TRAP,TRAZ,SIMP,SIMZ,FNCP,FNCZ
      FN=NX
      DEL=(FUL-FLL)/FN
      SSS=(0.,0.)
      ERR=0.01*ERRR/FN
      A=FLL
      DO 40 NNX=1,NX,1
      MXX=0
      B=A+DEL
      SS=(0.,0.)
      MX=2
      DX=DEL/2.
      LX=1
      X=A
      GO TO 15
 5    TRAZ=DX*SS
      MX=1
      LX=1
      DX=DEL
 10   SS=0.
      LX=LX+1
      DX=0.5*DX
      X=A+DX
 15   DO 20 IX=1,MX,1
      FSS=FCTI(X)
      SS=SS+FSS
 20   X=X+2.*DX
      IF(LX.EQ.I) GO TO 5
      MX=2*MX
      TRAP=0.5*TRAZ+DX*SS
      DIF=CABS(TRAP-TRAZ)
      IF(DIF.GE.DIP) MXX=MXX+1
      DIP=DIF
      SIMP=(4.*TRAP-TRAZ)/3.
      FNCP=(16.*SIMP-SIMZ)/15.
      ER=CABS(1.-FNCZ/FNCP)
      TRAZ=TRAP
      SIMZ=SIMP
      FNCZ=FNCP
      IF(LX.LT.4) GO TO 10
      IF(MXX.GT.4) GO TO 30
      IF(ER.GT.ERR) GO TO 10
 30   SSS=SSS+FNCP
 40   A=A+DEL
 50   CONTINUE
      RETURN
      END
$IBFTC FFNRM.   LIST
      SUBROUTINE FNORM(FNVX,FNVY,FNVZ,R,THT,PHI)
C     INPUT R,THT,PHI
      COMMON AR1,AR3,BR1,AR9,AR10,AR11,WAVE
      ST=SIN(THT)
      CT=COS(THT)
```

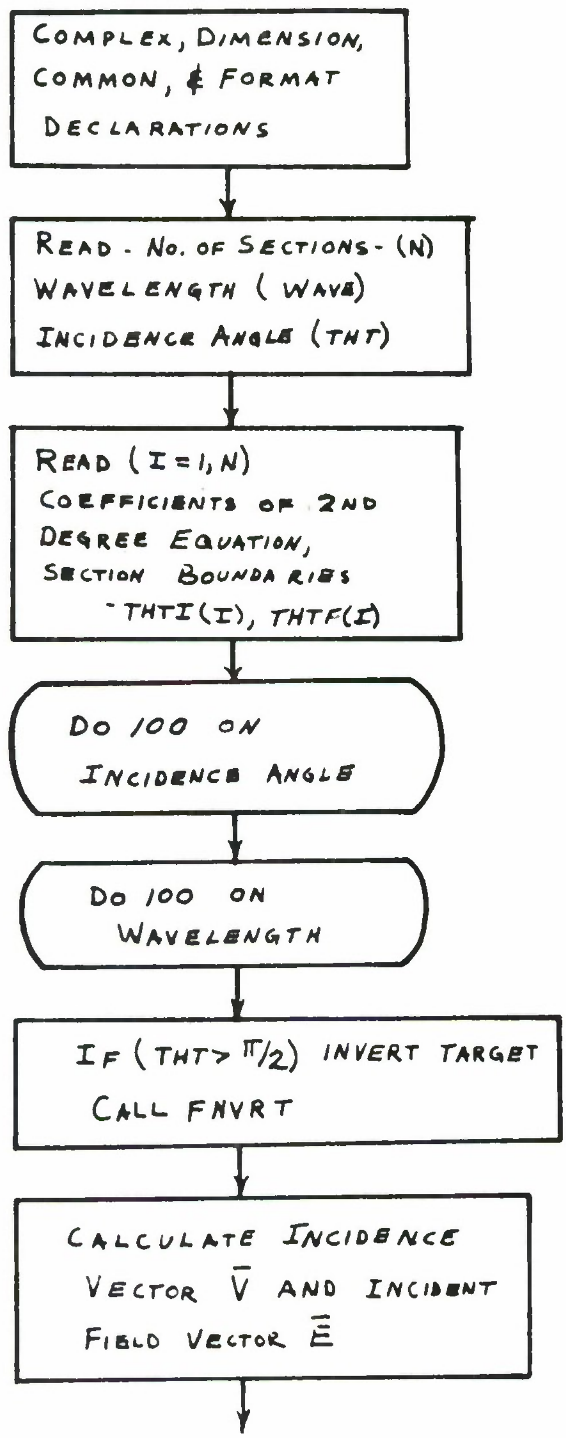
```
      SP=SIN(PHI)
      CP=COS(PHI)
      U=AR1*ST*ST+AR3*CT*CT+BR1*ST*CT
      V=AR9*CT+AR10*ST
      UTH=2.*(AR1-AR3)*ST*CT+BR1*(CT*CT-ST*ST)
       VTH=-AR9*ST+AR10*CT
      F1=2.*R*U+V
      F2=R*UTH+VTH
      FX=ST*CP*F1+CT*CP*F2
      FY=ST*SP*F1+CT*SP*F2
      FZ=CT*F1-ST*F2
      FN=SQRT(FX*FX+FY*FY+FZ*FZ)
      FNVX=FX/FN
      FNVY=FY/FN
      FNVZ=FZ/FN
      RETURN
      END
$IBFTC RADD     LIST
      FUNCTION RAD(THT)
C     CALCULATE R
      COMMON AR1,AR3,BR1,AR9,AR10,AR11,WAVE
      ST=SIN(THT)
      CT=COS(THT)
      C1=AR1*ST*ST+AR3*CT*CT+BR1*ST*CT
      C2=AR9*CT+AR10*ST
      C3=AR11
      IF(C1.EQ.0.)GO TO 11
      ARG=SQRT(C2*C2-4.*C1*C3)
      R=(-C2+ARG)/(2.*C1)
      R2=(-C2-ARG)/(2.*C1)
      IF(R2.LT.R.AND.R2.GT.0.) R=R2
      GO TO 12
 11   R=-C3/C2
 12   RAD=R
      RETURN
      END
$IBFTC FCOM.    LIST
      SUBROUTINE FCOMM(I)
      COMMONRA1,RA3,RB1,RA9,RA10,RA11,WAVE/DATA/AR1(20),AR3(20),BR1(20),
     CAR9(20),AR10(20),AR11(20)
      RA1=AR1(1)
      RA3=AR3(I)
      RB1=BR1(1)
      RA9=AR9(1)
      RA10=AR10(1)
      RA11=AR11(I)
      RETURN
      END
$DATA
```

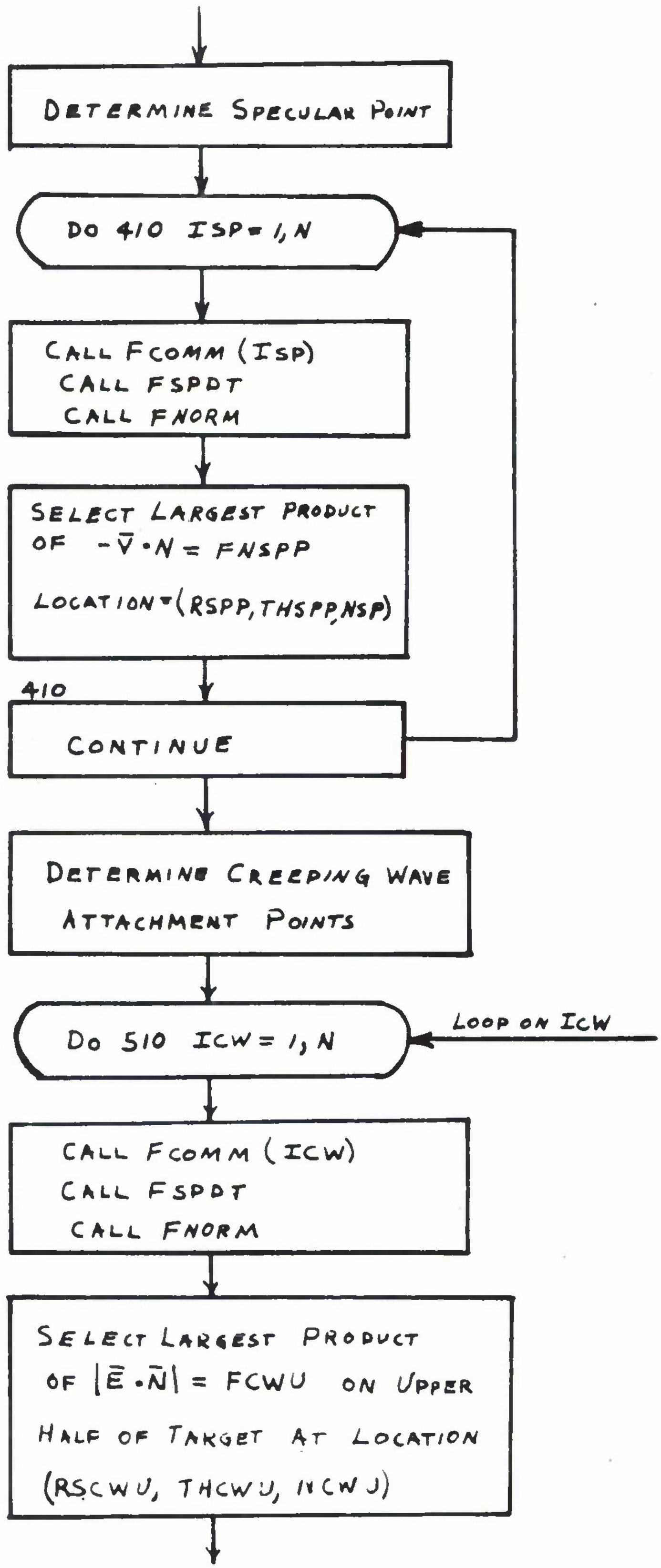
APPENDIX II
THE COMPUTER FLOW DIAGRAM

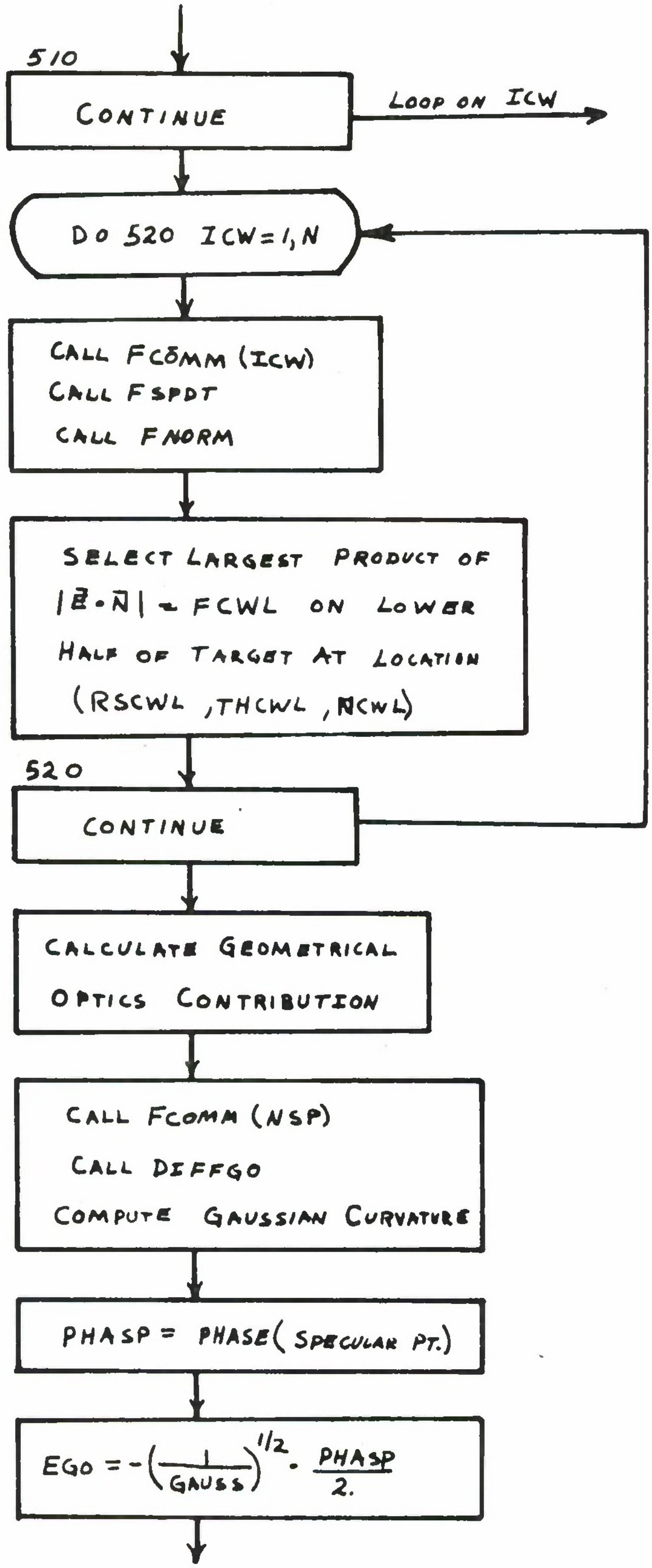
The flow diagram presented in Ref. 1 is presented here. This diagram shows the sequence in which the operations are performed.

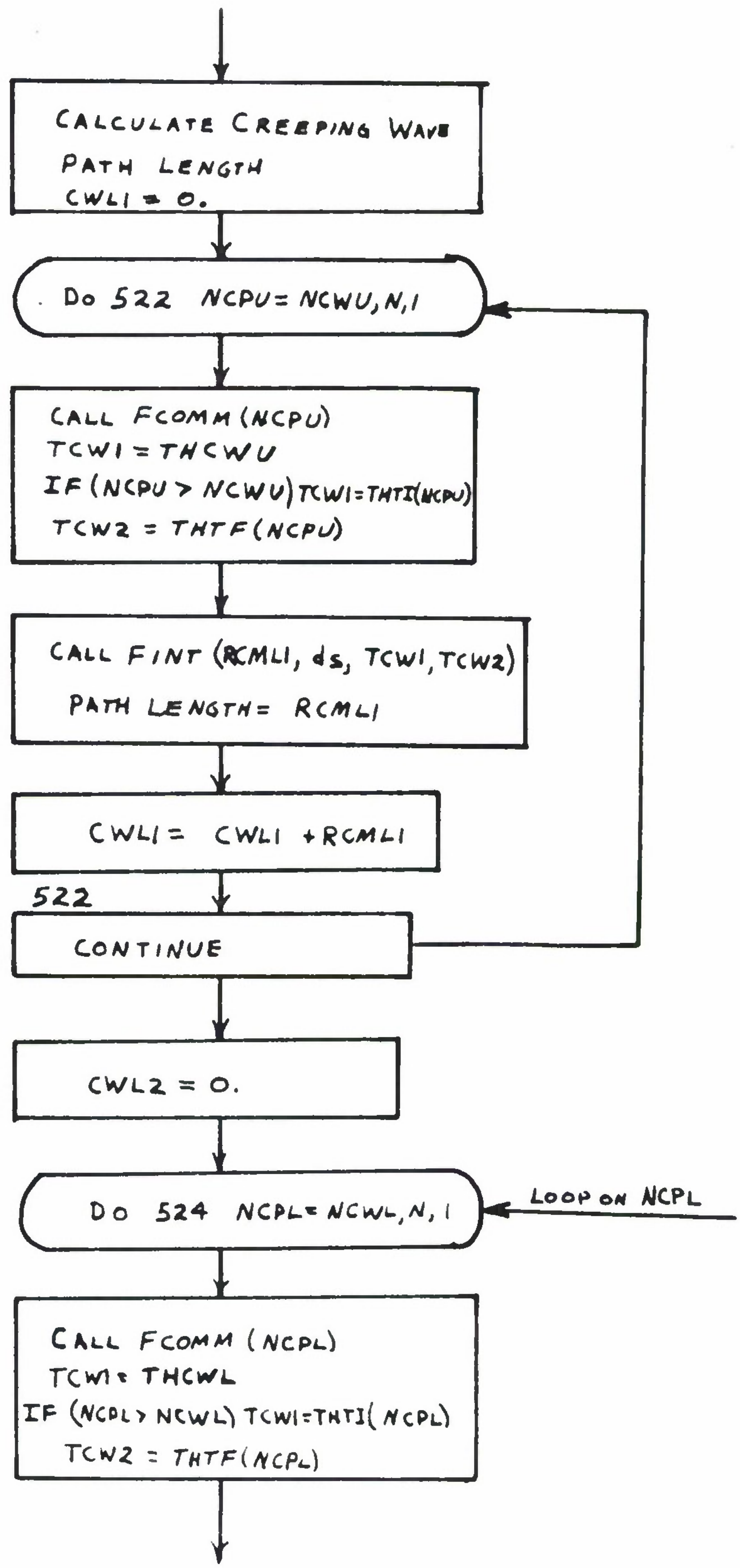
CREEPING WAVE COMPUTER PROGRAM

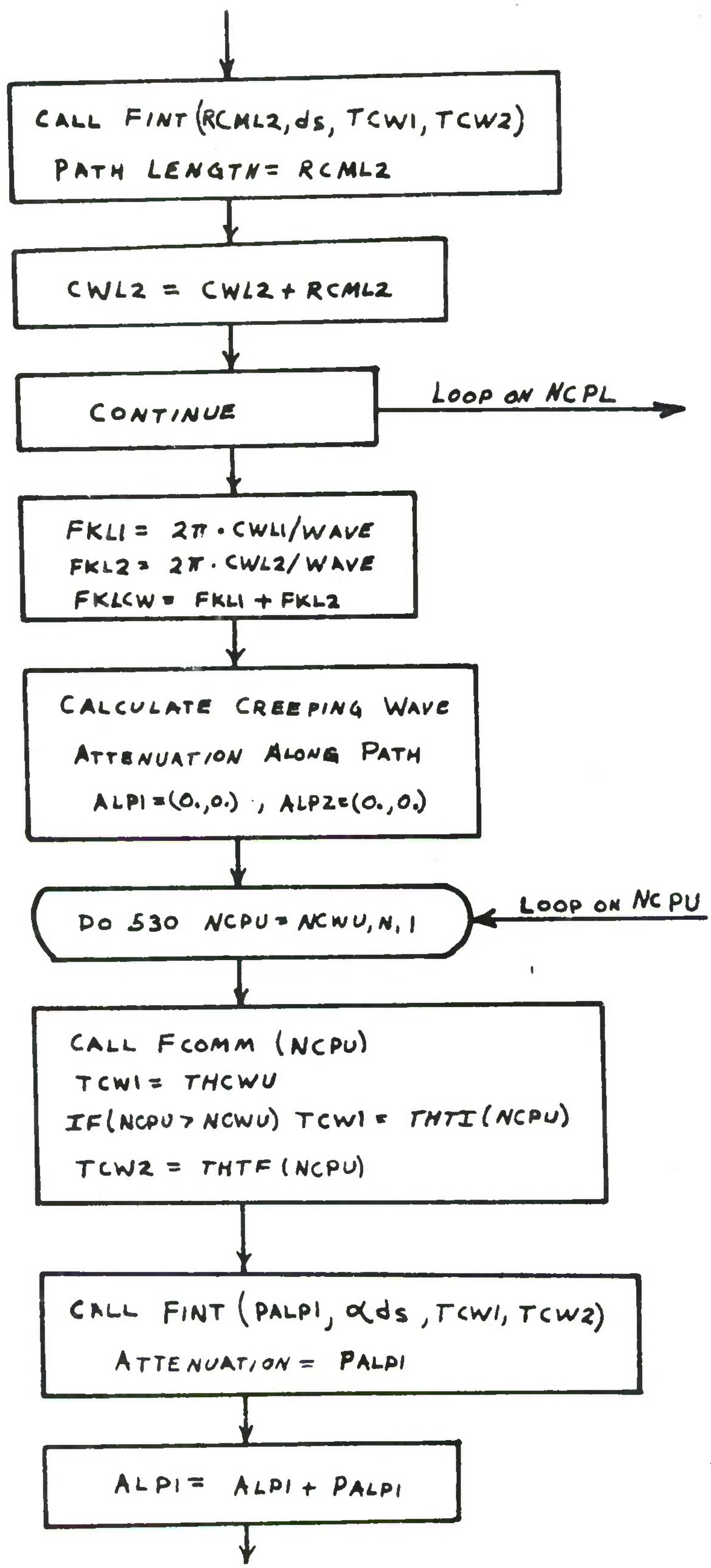
HORIZONTAL POLARIZATION

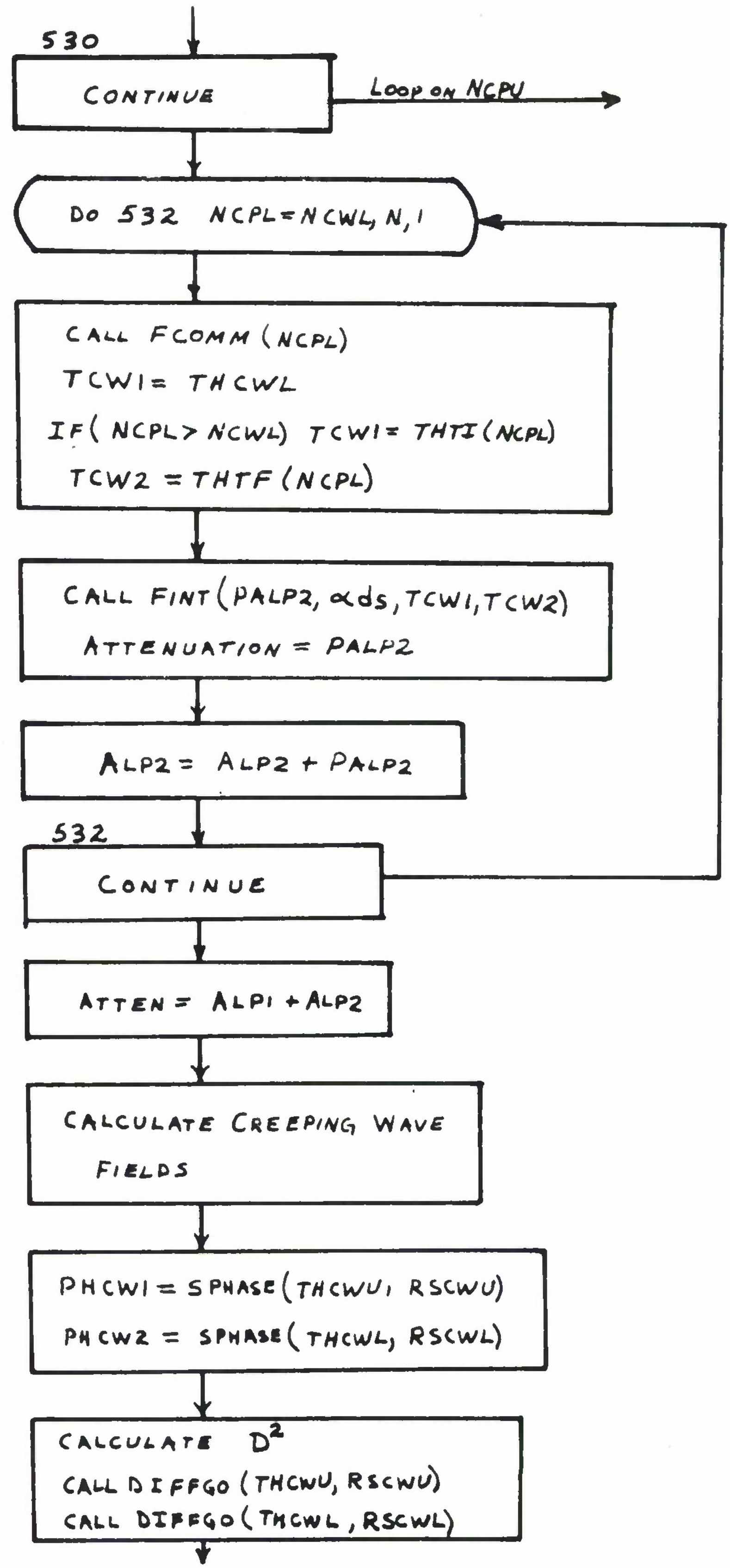


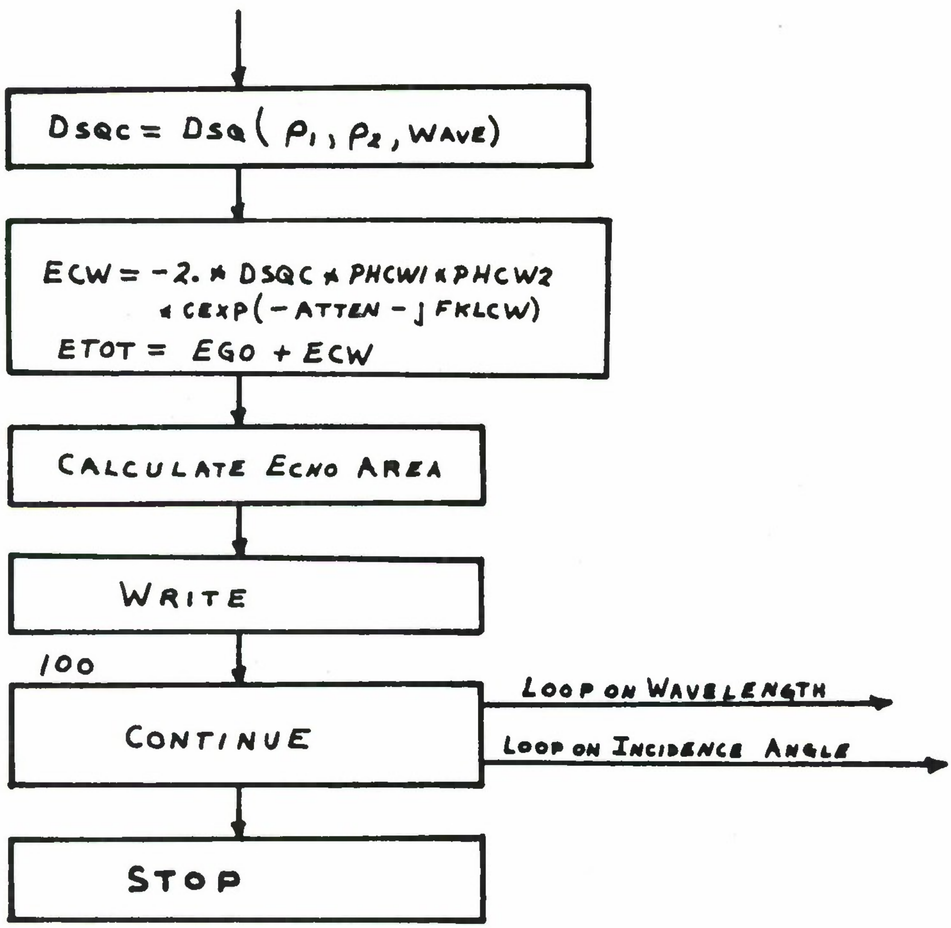

DETERMINE SPECULAR POINT
DO 410 ISP = 1, N
CALL FCOMM (ISP)
CALL FSPDT
CALL FNORM
SELECT LARGEST PRODUCT
OF $-\bar{V} \cdot N$ = FNSPP
LOCATION = (RSPP, THSPP, NSP)
410
CONTINUE
DETERMINE CREEPING WAVE
ATTACHMENT POINTS
DO 510 ICW = 1, N
LOOP ON ICW
CALL FCOMM (ICW)
CALL FSPDT
CALL FNORM
SELECT LARGEST PRODUCT
OF $|\bar{E} \cdot \bar{N}|$ = FCWU ON UPPER
HALF OF TARGET AT LOCATION
(RSCWU, THCWU, NCWU)


510
CONTINUE
LOOP ON ICW
DO 520 ICW=1,N
CALL FCŌMM (ICW)
CALL FSPDT
CALL FNORM
SELECT LARGEST PRODUCT OF
|E·N| ≈ FCWL ON LOWER
HALF OF TARGET AT LOCATION
(RSCWL ,THCWL ,NCWL)
520
CONTINUE
CALCULATE GEOMETRICAL
OPTICS CONTRIBUTION
CALL FCOMM (NSP)
CALL DIFFGO
COMPUTE GAUSSIAN CURVATURE
PHASP = PHASE(SPECULAR PT.)
EGO = -(1/GAUSS)^1/2 · PHASP/2.


CALCULATE CREEPING WAVE
PATH LENGTH
CWL1 = 0.
DO 522 NCPU = NCWU, N, 1
CALL FCOMM (NCPU)
TCW1 = THCWU
IF (NCPU > NCWU) TCW1 = THTI(NCPU)
TCW2 = THTF(NCPU)
CALL FINT (RCML1, ds, TCW1, TCW2)
PATH LENGTH = RCML1
CWL1 = CWL1 + RCML1
522
CONTINUE
CWL2 = 0.
DO 524 NCPL = NCWL, N, 1
LOOP ON NCPL
CALL FCOMM (NCPL)
TCW1 = THCWL
IF (NCPL > NCWL) TCW1 = THTI(NCPL)
TCW2 = THTF(NCPL)


CALL FINT (RCML2, ds, TCW1, TCW2)
PATH LENGTH = RCML2
CWL2 = CWL2 + RCML2
CONTINUE
LOOP ON NCPL
FKL1 = 2π · CWL1/WAVE
FKL2 = 2π · CWL2/WAVE
FKLCW = FKL1 + FKL2
CALCULATE CREEPING WAVE
ATTENUATION ALONG PATH
ALP1 = (0.,0.) , ALP2 = (0.,0.)
DO 530 NCPU = NCWU, N, 1
LOOP ON NCPU
CALL FCOMM (NCPU)
TCW1 = THCWU
IF(NCPU > NCWU) TCW1 = THTI(NCPU)
TCW2 = THTF (NCPU)
CALL FINT (PALP1, αds, TCW1, TCW2)
ATTENUATION = PALP1
ALP1 = ALP1 + PALP1


530
CONTINUE
LOOP ON NCPU
DO 532 NCPL=NCWL, N, 1
CALL FCOMM (NCPL)
TCW1= THCWL
IF(NCPL> NCWL) TCW1= THTI (NCPL)
TCW2 = THTF (NCPL)
CALL FINT (PALP2, αds, TCW1, TCW2)
ATTENUATION = PALP2
ALP2 = ALP2 + PALP2
532
CONTINUE
ATTEN = ALP1 + ALP2
CALCULATE CREEPING WAVE
FIELDS
PHCW1 = SPHASE (THCWU, RSCWU)
PHCW2 = SPHASE (THCWL, RSCWL)
CALCULATE D²
CALL DIFFGO (THCWU, RSCWU)
CALL DIFFGO (THCWL, RSCWL)


DSQC = DSQ(ρ1 , ρ2 , WAVE)
ECW = -2. * DSQC * PHCW1 * PHCW2
* CEXP(-ATTEN - j FKLCW)
ETOT = EGO + ECW
CALCULATE ECHO AREA
WRITE
100
CONTINUE
LOOP ON WAVELENGTH
LOOP ON INCIDENCE ANGLE
STOP

REFERENCES

1. Ryan, C.E., Jr., "A Geometrical Theory of Diffraction Analysis of the Radar Cross Section of a Sectionally Continuous Second Degree Surface of Revolution," Report 2430-4, April 1968, ElectroScience Laboratory, The Ohio State University Research Foundation; prepared under Contract F-19628-67-C-0318 for Department of the Air Force, Electronic Systems Division, L.G. Hanscom Field, Bedford, Massachusetts.

2. Ryan, C.E., Jr., "Memorandum on Analysis of Echo Area of Targets Using Geometrical Theory of Diffraction and Creeping Wave Theory," Report 2430-1, 22 May 1967, ElectroScience Laboratory, The Ohio State University Research Foundation; prepared under Contract F-19628-67-C-0318 for Department of the Air Force, Electronic Systems Division, L.C. Hanscom Field, Bedford, Massachusetts. (AD 658 469).

3. Richmond, J.H., "Scattering by Wire Loops and Square Plates in the Resonance Region," Report 2097-1, 28 January 1966, Antenna Laboratory, The Ohio State University Research Foundation; prepared under Contract No. AF 19(628)-4883 for Air Force Systems Command, United States Air Force, Bedford, Massachusetts.

4. Ryan, C.E., Jr., "A Computer Program for Backscatter by Targets Composed of Cones, Cylinders, and Disks," (in process), ElectroScience Laboratory, The Ohio State University Research Foundation; prepared under Contract F-19628-67-C-0318 for Department of the Air Force, Electronic Systems Division, L.C. Hanscom Field, Bedford, Massachusetts.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* ElectroScience Laboratory The Ohio State University, Department of Electrical Engineering, 1320 Kinnear Rd., Columbus, Ohio	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP N/A

3. REPORT TITLE
A Computer Program for Backscatter by Smoothly Joined, Second Degree Surfaces of Revolution

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Technical Report

5. AUTHOR(S) *(Last name, first name, initial)*
Ryan, C.E., Jr.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1968	25	4

8a. CONTRACT OR GRANT NO. F-19628-67-C-0308	9a. ORIGINATOR'S REPORT NUMBER(S) ESD-TR-68-213
b. PROJECT NO.	
c. TASK	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. AVAILABILITY/LIMITATION NOTICES
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Deputy for Surveillance and Control Systems, Electronic Systems Division L.G. Hanscom Field, Bedford, Mass.

13. ABSTRACT

A computer program for calculation of the echo area of smoothly joined, N section convex conducting surfaces, described by a second degree equation is described. For the case of $E\theta$ (parallel) polarization of the incident and scattered fields the solution is obtained by a combination of geometrical optics and creeping wave theory. For the case of $E\phi$ (perpendicular) polarization the solution is obtained using geometrical optics, and the creeping wave is neglected. The computed results for $E\theta$ polarization are in good agreement with measurements for prolate spheroids, prolate spheroid-sphere, and prolate spheroid-oblate spheroid combinations.

DD FORM 1 JAN 64 1473

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Radar Cross Section						
Backscatter						
Surface of Revolution						
Computer Program						
Geometrical Optics						
Creeping Waves						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate outhor)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
_______________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
_______________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
_______________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.